International Journal of Research in Advent Technology, Vol.2, No.3, March 2014
E-ISSN: 2321-9637

Criminisis's Algorithm

Ms. Kamble Shubhangi B. [1]
Electronics and Telecommunicationt[1], Rajendra Mane College of Engg & Tech.Ambav[1]
Email:shubs.83@rediffmail.com[1]

Abstract-A new algorithm is proposed for removing large objects from digital images. The challenge is to fill in the hole that is left behind in a visually plausible way. In the past, this problem has been addressed by two classes of algorithms: (i) "texture synthesis" algorithms for generating large image regions from sample textures, and (ii) "inpainting" techniques for filling in small image gaps This paper presents a novel and efficient algorithm that combines the advantages of these two approaches

Index Terms- texture synthesis**;**inpainting**;** object removal.

1. INTRODUCTION

New algorithm is proposed for removing large objects from digital images. The challenge is to fill in the hole that is left behind in a visually plausible way.

Figure1 an example of this task, where the building (manually selected as the target region) is automatically replaced by data sampled from the remainder of the image

Fig. 1Shows Removing large objects from images. (a) Original photograph. (b) The region corresponding to the building (covering about 19% of the image) has been manually selected and then automatically removed. Notice that the all structures of the building have been synthesized in the occluded area together with the grass and rock textures .

2. PRESENT THEORY AND PRACTICES

In the past, this problem has been addressed by two classes of algorithms: (i) "texture synthesis" algorithms for generating large image regions from sample textures, and (ii) "inpainting" techniques for filling in small image gaps. The former work well for "textures" -- repeating two-dimensional patterns with some stochastic; the latter focus on linear "structures" which can be thought of as one-dimensional patterns, such as lines and object contours

.

a) ***b)***

Figure.2 Removing large objects from photographs. a) Original image b) The result of region filling by traditional image inpainting.

Notice the blur introduced by the diffusion process and the complete lack of texture in the synthesized area c) The final image where the bungee jumper has been completely removed and the occluded region reconstructed by our automatic algorithm

3. KEY OBSERVATIONS

3.1 Exemplar-based synthesis suffices

The core of our algorithm is an isophote-driven image sampling process. It is well-understood that exemplar-based approaches perform well for two-dimensional textures [1], [11],[17]. But, we note in addition that exemplar-based texture synthesis is sufficient for propagating extended linear image structures, as well; i.e., a separate synthesis mechanism is not required for handling isophotes.

The core of our algorithm is an isophote-driven image sampling process. It is well-understood that exemplar-based approaches perform well for two-dimensional textures [1], [11],[17]. But, we note in addition that exemplar-based texture synthesis is sufficient for propagating extended linear image structures, as well; i.e., a separate synthesis mechanism is not required for handling isophotes.

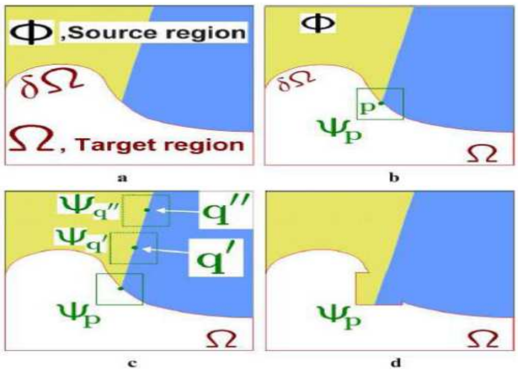

Figure 3.1 illustrates this point. For ease of comparison, we adopt notation similar to that used in the inpainting literature

The region to be filled, i.e., the target region is indicated by Ω, and its contour is denoted δΩ. The contour evolves inward as the algorithm progresses, and so we also refer to it as the ―fill front‖. The source region, Φ which remains fixed throughout the algorithm, provides samples used in the filling process.
Fig.3.1 Structure propagation by exemplar-based texture syn-thesis. (a) Original image, with the target region Ω, its contour δΩ, and the source region Φ clearly marked. (b) We want to synthesize the area delimited by the patch cantered on the point p ε ∂Ω. (c) The most likely candidate matches for lie along the boundary between the two textures in the source region e.g. and (d) The best matching patch in the candidates set has been copied into the position occupied by , thus achieving partial filling of . Notice that both texture and structure (the separating line) have been propagated inside the target region. The target region Ω has, now, shrunk and its front δΩ has assumed a different shape.
the pixels from Ψq to fill Ψp. With a new contour, the next round of finding the patch with the highest continues, until all the gaps are filled.

3.2 Filling order is critical.

The previous section has shown how careful exemplar-based filling may be capable of propagating both texture and struc-ture information. This section demonstrates that the quality of the output image synthesis is highly influenced by the order in which the filling process proceeds. Furthermore, we list a number of desired properties of the ―ideal‖ filling algorithm.
A comparison between the standard concentric layer filling (onion-peel) and the desired filling behaviour is illustrated in fig. 3.2 Figures 3b,c,d show the progressive filling of a concave target region via an anti-clockwise onion-peel strategy. As it can be observed, this ordering of the filled patches produces the horizontal boundary between the background image regions to be unexpectedly reconstructed as a curve.
The user will be asked to select a target region, Ω, manually. (a) The contour of the target region is denoted as δΩ. (b) For every point p on the contour δΩ, a patch Ψp is constructed, with p in the centre of the patch. A priority is calculated based on how much reliable information around the pixel, as well as the isophote at this point. (c) The patch with the highest priori-ty would be the target to fill. A global search is performed on the whole image to find a patch, Ψq that has most similarities with Ψp. (d) The last step would be copy

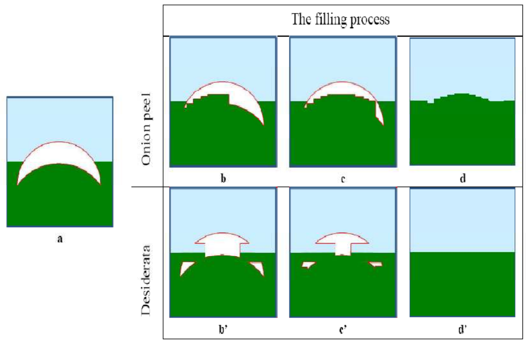

Fig. 3.2 The importance of the filling order when dealing with concave target regions. (a) A diagram showing an image and a selected target region (in white). The remainder of the image is the source. (b,c,d) Different stages in the concentric-layer fil-li ng of the target region. (d) The onion-peel approach pro-duces artifacts in the synthesized horizontal structure. (b‘, c‘, d‘) Filling the target region by an edge-driven filling order achieves the desired artifact-free reconstruction. (d‘) The final edge-driven reconstruction, where the boundary between the two background image regions has been reconstructed correctly.

4. OUR REGION-FILLING ALGORITHM

First, given an input image, the user selects a target region, Ω, to be removed and filled. The source region, may be defined as the entire image minus the target region (Φ = I-Ω,), as a dilated band around the target region, or it may be manually specified by the user.
Algorithm iterates the following three steps until all pixels have been filled.

- Computing Patch Priorities.
- Propagating Texture and Structure Information.
- Updating Confidence Values.
- Computing Patch Priorities.

The priority computation is biased toward those patches which:

a) Are on the continuation of strong edges.
b) Are surrounded by high-confidence pixels. Given a patch centered at the point ‗p‘.We defines its priority as the product of two terms

P(P)=C(P)D(P) (1)

equation (1) is used to calculate patch priorities.

C(P)The confidence term that measure of the amount of reliable information surrounding the pixel 'p'

$$c(p)= \frac{\sum[(q\in{}^{\varphi}p)\cap(I-\Omega)]c(p)}{|\Psi p|} \qquad (2)$$

equation (2) is used to calculating confidence term.

D(P)The data term that is a function of the strength of iso-photes hitting the front $\partial\Omega$ at each iteration which is find out by equation(3) .

$$D(p)=\frac{\left|\nabla I_{p}^{\perp}\ \eta_p\right|}{\alpha} \qquad (3)$$

Where, (a) η_p estimated as the unit vector orthogonal to Ψp the front $\partial\Omega$..(b) $\nabla I_{p}^{\perp}$. Is the isophote (direction and intensity) at point ‘p’ It computed as the maximum value of the image gradient in $\Psi p \cap I$ (c) α=Normalization factor (e.g. For Gray level it is α=255)

.Normalization factor (e.g. For Gray level it is α=255)

- Propagating Texture and Structure Information.

Propagate image texture by direct sampling of the source re-gion, the distance $\partial({}^{\varphi}p \quad {}^{\varphi}q)$ between two generic patches ${}^{\varphi}p$ and ${}^{\varphi}q$ is simply defined as the sum of squared differences is shown in equation(4)

$${}^{\varphi}q = \arg\min_{{}^{\varphi}_{q} \in \Phi} d\left({}^{\varphi}_{p}, {}^{\varphi}_{q}\right) \qquad (4)$$

- Updating Confidence Values.

After the patch ${}^{\varphi}_{p}$ has been filled with new pixel values, the confidence c(p) is updated in the area delimited by ${}^{\varphi}_{p}$ which is shown in equation (5)

$$c(p)=c(\hat{p}) \qquad \forall_p \square ({}^{\varphi}p \ \cap T) \qquad (5)$$

5. SOME PROPERTIES OF CRIMINISIS'S ALGORITHM

The effect of the confidence term is that of smoothing the contour of the target region by removing sharp appendices and making the target contour close to circular. . Notice that (1) only dictates the order in which filling happens. The use of image patches for the actual filling achieves texture synthesis [11]. Furthermore, since the fill order of the target region is dictated solely by the priority function P(p), we avoid having to predefine an arbitrary fill order as done in existing patch-based approaches [11], Our fill order is function of image properties, resulting in an organic synthesis process that eliminates the risk of ―broken-structure‖ artifacts (as in fig. 11f). Furthermore, since the gradient-based guidance tends to propagate strong edges, blocky and misalignment artifacts are reduced (though not completely eliminated), without a patch-cutting (quilting) step [11] or a blur-inducing blending step. It must be stressed that our algorithm does not use explicit nor implicit segmentation at any stage. For instance, the gradient operator in (1) is never thresholded and real valued numbers are employed.

6. IMPLEMENTATION DETAILS

In our implementation the contour $\partial\Omega$ of the target region is modelled as a dense list of image point locations. These points are interactively selected by the user via a simple drawing interface. Given a point $p \in \partial\Omega$, the normal direction np is computed as follows: i) the positions of the ―control‖ points of $\partial\Omega$ are filtered via a bi-dimensional Gaussian kernel and, ii) np is estimated as the unit vector orthogonal to the line through the preceding and the successive points in the list. Alternative implementation may make use of curve model fitting. The gradient is computed as the maximum value of the image gradient in. Robust filtering techniques may also be employed here. Finally, pixels are classified as belonging to the target region, the source region or the remainder of the image by assigning different values to their alpha compo-nent. The image alpha channel is, therefore, updated (locally) at each iteration of the filling algorithm.

7. RESULT AND COMPARISIONS

Here we apply our algorithm to a variety of images, ranging from purely synthetic images to full-color photographs that include complex textures. Where possible, we make side-by-side comparisons to previously proposed methods. In other cases, we hope the reader will refer to the original source of our test images (many are taken from previous literature on inpainting and texture synthesis) and compare these results with the results of earlier work. All experiments were run on a 2.5GHz Pentium IV with 1GB of RAM.

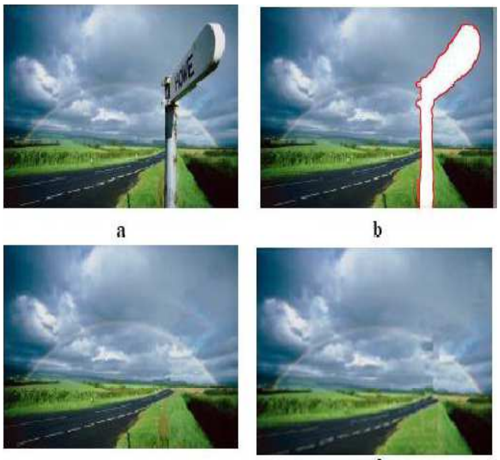

Fig. 7.1 Comparisons with Jia et al. ―Image Repairing‖ [16]. (a) Original input image. (b) The manually selected target region. (c) The resulting region-filling achieved by Jia et al. (d) the result of our region-filling algorithm. The missing portion of rainbow is reconstructed convincingly. Figures (c) and (d) are of comparable quality, but our algorithm avoids the image segmentation step with considerable increase in speed..

a) b) c)

Fig. 7.2 Removing the multiple objects from photographs. (a) Original photograph of Cricket. (b) inpainted image.,(c)) Several ob-jects are sequentially removed. Notice how well the structures have been reconstructed.

8. CONCLUSION AND FUTURE WORK

This paper has presented a novel algorithm for removing large objects from digital photographs. The result is an image in which the selected object has been replaced by a visually plausible background that mimics the appearance of the source region.

Our approach employs an exemplar-based texture synthesis technique modulated by a unified scheme for determining the fill order of the target region. Pixels maintain a confidence value, which together with image isophotes, influence their fill priority.

The technique is capable of propagating both linear structure and two-dimensional texture into the target region with a single, simple algorithm. Comparative experiments show that a simple selection of the fill order is necessary and sufficient to handle this task.

Advantages

a) Preservation of edge sharpness
b) No dependency on image segmentation

Disadvantages

a) The synthesis of regions for which similar patches do not exist does not produce reasonable results (a problem common to [10], [19]);
b) The algorithm is not designed to handle curved structures

Currently we are investigating extensions of the current algorithm to handle accurate propagation of curved structures in still photographs as well as removing objects from video, which promise to impose an entirely new set of challenges.

REFERENCES

[1] A.Criminisi, P.Perez and K. Toyama. "Region Filling and Object Removal by Exemplar-Based Image inpainting" IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 13, NO. 9,SEP 2004

[2] P. Harrison. "A non-hierarchical procedure for re-synthesis of complex texture." In Proc. Int. Conf. Central Europe Comp. Graphics, Visua. and Comp. Vision, Plzen, Czech Republic, February 2001Bharat

[3] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. "Simultaneous structure and texture image inpainting." In Proc. Conf. Comp. Vision Pattern Rec., Madison, WI, 2003.

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. "Image inpainting." In Proc. ACM Conf. Comp. Graphics (SIGGRAPH), pages 417–424, New Orleans, LU, July 2000